

APPLICATION FOR UNITED STATES LETTERS PATENT

For

**Automatic Phase Lock Loop Design Using Geometric Programming**

Inventors:

David M. Colleran

Arrash Hassibi

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP  
12400 Wilshire Boulevard  
Los Angeles, CA 90025-1026  
(408) 720-8300

Attorney's Docket No.: 004363.P005C

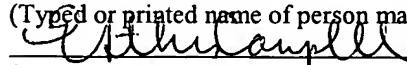
"Express Mail" mailing label number: EV 409359896 US

Date of Deposit: March 26, 2004

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to Mailstop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Esther Campbell

(Typed or printed name of person mailing paper or fee)

  
(Signature of person mailing paper or fee)

March 26, 2004

(Date signed)

## **Automatic Phase Lock Loop Design Using Geometric Programming**

### **Related Application**

[0001] This application is a continuation of the prior application for “Automatic Phase Lock Loop Design Using Geometric Programming” filed by David M. Colleran et al. on April 7, 2002 (U.S. Patent Application Serial No. 10/119,347).

### **Field of Invention**

[0002] The field of invention relates generally to circuit design; and, more specifically, to automatic phase lock loop design using geometric programming.

## **BACKGROUND**

### **1.0 Phase Lock Loop (PLL) Theory of Operation**

[0003] Figure 1 provides an illustration of a Phase-Lock-Loop (PLL) 100. As seen in Figure 1, a typical PLL includes a phase detector 101, a charge pump 102, a loop filter 103, a Voltage Controlled Oscillator (VCO) 104 and a feedback divider 105. PLLs are often used for frequency multiplication, wherein, the frequency of an input reference signal 106 is effectively multiplied. That is, the output signal 107 frequency of the PLL 100 (which corresponds to the output signal 107 frequency of the VCO 104) will be a multiple of the frequency of the input reference signal 106. According to basic principles of PLL operation, the multiplication is coextensive with the frequency division “M” performed by feedback divider 105.

[0004] Better said, if the feedback divider 105 reduces the frequency of the VCO output signal 107 by a factor of M, the frequency of the VCO output signal 107 should be a factor M of the reference signal 106 frequency. A working PLL 100 may be thought of as a stable control system that is designed to “automatically” drive an error term magnitude to zero. Here, the error term magnitude can be thought of as effectively representing the difference between the actual VCO output signal 107 frequency and the correct VCO output signal 107 frequency. In reality, the error term magnitude represents the difference between the actual VCO output signal 107 phase and the correct VCO output signal 107 phase. The correct VCO output signal 107 frequency, as discussed above, corresponds to  $Mf_{REF}$  where M is the frequency division performed by the feedback divider 105 and  $f_{REF}$  is the input reference signal 106 frequency.

[0005] Thus, when the VCO output signal 107 frequency deviates from  $Mf_{REF}$ , a non zero error term magnitude is created that causes the VCO output signal 107 frequency to change in a direction that corrects for the error. Notice that the error term may also be described as having a polarity (e.g., positive or negative) so that the appropriate direction of VCO output signal 107

frequency change is represented. For example, if the VCO output signal 107 frequency is less than  $Mf_{REF}$  (i.e., the VCO output signal 107 frequency is too low), a positive error term is created that causes the VCO output signal 107 frequency to increase. Likewise, if the VCO output signal 107 frequency is greater than  $Mf_{REF}$  (i.e., the VCO output signal 107 frequency is too high), a negative error term is created that causes the VCO output signal 107 frequency to decrease.

[0006] The polarity of the error term is a matter of definition and may be “flipped” from that described just above depending on the particular definitional preference of the designer (i.e., some designs may choose to define a negative error term as corresponding to a situation where the VCO output signal is too low and a positive error term as corresponding to a situation where the VCO output signal is too high). For convenience, the terminology initially described above (wherein a positive error term causes a VCO output signal 107 frequency increase and a negative error term causes a VCO output signal 107 frequency decrease) will be used throughout the present application.

[0007] Referring to the PLL topology 100 of Figure 1, the error term is presented at the output of the phase detector 101. As such, in order to effectively compare the “actual” VCO output signal 107 frequency against the “correct” VCO output signal 107 frequency (so that an appropriate error term may be crafted), notice that the phase detector 101 receives both the input reference signal 106 and the output signal 108 of the feedback divider 105. With a correct VCO output signal frequency corresponding to  $Mf_{REF}$ ; and, with the feedback divider 105 dividing down the frequency of the VCO output signal 107 by a factor of  $M$ ; note that, when the frequency of the VCO output signal 107 is correct (i.e., is equal to  $Mf_{REF}$ ), the frequency of the input reference signal 106 (i.e.,  $f_{REF}$ ) and the frequency of the frequency divider output signal

108 should be the same (i.e., the frequency of the feedback divider output signal 108 should also be  $f_{REF}$ ).

**[0008]** Likewise, deviations of the VCO output signal 107 frequency away from its correct value ( $Mf_{REF}$ ) would cause the frequency of the feedback divider output signal 108 to correspondingly deviate from a frequency of  $f_{REF}$ . As such, the phase detector 101 effectively uses the input reference signal 106 frequency as a “benchmark” against which the frequency of the feedback divider output signal 108 is compared. The comparison results in an error term being crafted having both magnitude information (which expresses the extent to which the pair of frequencies are dissimilar) and polarity information (which expresses which signal has greater frequency). When the frequencies are the same, the error term should have a magnitude of zero (meaning no change in VCO output signal 108 frequency is needed); and, when the frequencies are different, the error term should have a magnitude that is representative of the amount of difference as well as a polarity that indicates whether or not the feedback divider output signal 108 frequency is less than or greater than  $f_{REF}$ .

**[0009]** The phase detector 101 is typically designed to effectively compare the frequencies of its input signals 106, 108 in the time domain rather than the frequency domain; and, as such, compares the phase positioning of signals 106, 108 (hence the name “phase” detector). Here, as depicted at the inset 109 of Figure 1, if the phase detector 101 observes that an edge of the feedback divider output signal 108 is “lagging” an edge of the input reference signal 106, the phase detector will craft a positive error term (indicating that the VCO output signal 107 frequency is too low); and, likewise, as observed at inset 110, if the phase detector 101 observes that an edge of the feedback divider output signal 108 is “leading” 111 an edge of the input

reference signal 106, the phase detector will craft a negative error term (indicating the VCO output signal 107 frequency is too high).

[0010] In either case, the magnitude of the error term will indicate the extent to which the edges being compared are misaligned in time. If the VCO output signal frequency and phase are “correct”, the edges should be aligned in time (as indicated at inset 111) which corresponds to an error term having zero magnitude. Note that the particular phase detector 101 embodiment of Figure 1 uses different outputs to indicate error term polarity. That is, if the error term polarity is positive (so as to cause an increase in VCO output signal 107 frequency), a pulse will appear at a first output 112 (“up”); and, if the error term polarity is negative (so as to cause a decrease in VCO output signal 107 frequency), a pulse will appear at a second output 113 (“down”). In either case, the width of the pulse in time reflects the magnitude of the error term.

[0011] The PLL 100 is configured such that, after an initial period of time, the PLL will automatically acquire “phase lock” as depicted in depiction 111. This effectively corresponds to the PLL automatically driving the error term magnitude to zero (which also corresponds in kind to the VCO output signal 107 frequency being driven to a value of  $Mf_{REF}$ ) over this same initial time period. The error term magnitude is automatically driven to zero because of the activity of the charge pump 102, loop filter 103 and VCO 104. A VCO 104 is a circuit that produces, at its output, a signal 107 whose frequency is a function of the voltage placed at its input 114. The charge pump 102 and loop filter 103 effectively cause a change in the voltage level of VCO input 114, where the voltage change is related to the error term information. That is, the size of the voltage change (e.g., in volts) is related to the magnitude of the error term and the direction of the voltage change (e.g., up or down) is related to the polarity of the error term.

[0012] As the loop filter 103 is comprised of a capacitor that is shunted to ground, the voltage level of the VCO input 114 can be increased by pumping charge into the loop filter 103; and, the voltage level of the VCO input 114 can be decreased by extracting charge from the loop filter 103. The charge pump 102 serves this function as it is configured to: 1) pump charge into the loop filter 103 so as to raise the VCO input 114 voltage (e.g., in response to a positive error term); and, 2) extract charge from the loop filter 103 so as to lower the VCO input 114 voltage (e.g., in response to a negative error term). The amount of charge pumped into or extracted from the loop filter 103 is proportional to the magnitude of the error term (so that the voltage change is proportional to the magnitude of the error term).

[0013] The loop filter 103 “holds” the voltage level of the VCO input 114 to that which the charge pump 102 changed it to. Here, a voltage change that causes the VCO output signal 107 frequency to be more accurate (i.e., closer to  $Mf_{REF}$ ) should cause a subsequent error term to have less magnitude; while, a voltage change that causes the VCO output signal 107 frequency to be less accurate (i.e., farther from  $Mf_{REF}$ ) should cause a subsequent error term to have greater magnitude. Thus, the more accurate the VCO output signal 107 frequency becomes, the less the voltage level at the VCO input 114 will be changed. This activity causes the PLL 100 to naturally guide its VCO input 114 voltage level to a voltage that corresponds to the correct VCO output signal 107 frequency. This, in turn, corresponds to the error magnitude term being automatically driven to zero.

## 2.0 Phase Lock Loop Characteristics

[0014] The previous discussion provided an overview of a theory of operation for a PLL. PLLs are typically constructed with electronic circuit components (e.g., transistors, capacitors, resistors, etc.) that have limited performance and/or performance characteristics that may vary as

a function of manufacturing tolerances and/or environmental conditions. As such, when actually implemented in hardware, a PLL may stray from its perfect theoretical operation or otherwise possess imperfections that can affect other circuitry that depends on the operation of the PLL. Furthermore, the dynamics of “loop stability” (i.e., wherein an error term magnitude is automatically driven to zero) are more complicated than that described in the preceding section to the extent some additional “conditions” apply to the PLL’s basic building blocks 101, 102, 103, 104, 105. As such, various performance characteristics are understood to exist for PLLs that help describe the imperfections and/or help identify whether or not the PLL will be stable over the region of desired operation. Some of these include (among possible others): 1) static phase error; 2) stability; and 3) jitter. A discussion of each of these is provided immediately below.

## 2.1 Static Phase Error

[0015] From the discussion of Section 1.0 (and in particular to the discussion of phase lock described with respect to inset 112 of Figure 1), recall that a PLL should normally adjust the VCO output signal 107 frequency until the edges of the reference input signal 106 and the feedback divider output signal 108 are aligned with one another. By contrast, Figure 2A shows a depiction of the static phase error that may exist within a PLL. From the depiction of Figure 2A it is evident that static phase error corresponds to the tendency of a PLL to lock onto a mis-aligned relationship between the reference input signal 206 and the feedback divider output signal 208.

[0016] Typically, when a PLL exhibits static phase error, it is because of imperfections in the phase detector 101 and charge pump 102 that “effectively” cause a mis-aligned reference input signal 206/feedback divider output signal 208 condition (e.g., as depicted in Figure 2A) to



produce an error term having zero magnitude (i.e., no error term) from the perspective of the VCO input 114. Likewise, a properly aligned reference input signal/feedback divider output signal condition (e.g., as depicted at inset 111 of Figure 1), “effectively” causes an error term having non zero magnitude from the perspective of the VCO input 114. This, in turn, corresponds to the VCO output signal 107 phase being misaligned from the phase of the correct value ( $M_{fref}$ ), even though the VCO output signal 107 frequency is the correct value ( $M_{fref}$ ).

## 2.2 Stability

[0017] Stability is a term of art used to refer to a pair of characteristics that can be used to describe whether or not a PLL will automatically drive the error term magnitude to zero (so as to produce the correct VCO output signal frequency). In general, the stability of a PLL depends upon whether or not it is behaving as a negative feedback loop. The PLL theory of operation provided in Section 1.0 effectively described negative feedback because (as described in Section 1.0): 1) if the frequency of the VCO output signal 107 was greater than its “correct” value of  $M_{fREF}$ , the PLL acted to reduce the VCO output signal 107 frequency; and 2) if the frequency of the VCO output signal 107 was less than its “correct” value of  $M_{fREF}$ , the PLL acted to increase the VCO output signal 107 frequency.

[0018] If the PLL were to operate inapposite to that described just above (i.e., if the VCO output signal frequency was increased in response to the phase detector’s determining that the VCO output signal was above its correct output frequency; and/or, if the VCO output signal frequency was decreased in response to the phase detector’s determining that the VCO output signal was beneath its correct output frequency), the PLL would be unstable because the VCO output signal frequency would tend to continually rise or tend to continually fall. Typically, the

PLL parameters of “cross-over-frequency” and “phase margin” are used to help understand whether the PLL is operating within the realm of negative feedback.

**[0019]** A depiction of the cross over frequency 210 and phase margin 211 are shown in Figure 2B. The cross over frequency is the frequency at which the transfer function 212 of the PLL reaches unity. Here, as the loop filter 103 is a low pass filter, the PLL channel (which can be viewed as the signal processing that occurs through one pass through the loop of the PLL) has a response that varies with frequency as depicted in transfer function 212. The phase margin 211 of the PLL is the difference between the phase delay through the PLL channel and 180 degrees at the cross over frequency 210. An example of the difference between the phase delay and 180 degrees of the PLL channel is observed in curve 213 of Figure 2B. Here, the more the phase margin rises above -180 degrees, the more the PLL is operating within a negative feedback (and therefore stable) region of operation.

### 2.3 Jitter

**[0020]** Jitter relates to the variation in the period of the VCO output signal 107 with respect to an ideal output signal,  $M_{fref}$ , once phase lock has been acquired. Because of systematic variations at the VCO input 114, as shown in Figure 2C, as well as random variations (due to thermal noise, for example), the instantaneous frequency at the VCO output signal 107 will not always be equal to  $M_{fref}$ , even though the average VCO output signal 107 frequency is correct. This will cause the time at which the rising edge of the VCO output signal 107 occurs to vary with respect to where the rising edge of  $M_{fref}$  would ideally occur. The maximum variation of the rising edges is undesirable, and is called jitter.

### 3.0 Lack of Widely Accepted Automated Analog and/or Mixed Signal Design Technology

[0021] Digital signal processing “reacts” to an input signal depending on whether the input signal is recognized as a “1” or a “0”. Likewise, digital signal processing generates an output signal from the perspective that the output signal corresponds to a “1” or a “0”. Analog signal processing, by contrast, views an input signal as being continuous (rather than discrete “1”s or “0”s); and, on the output side, tailors a continuous output signal (as opposed to discrete “1”s or “0”s). When both analog and digital signal processing techniques are significantly involved in the overall function of an electronic circuit, the electronic circuit may be referred to as a “mixed signal” circuit.

[0022] That is, mixed signal refers to the notion that both digital signal processing and analog signal processing are being performed with the same circuit. As such, mixed signal circuitry and analog circuitry are both configured to process analog signals (and, likewise, contain some degree of analog circuitry). Here, PLLs are often characterized as mixed signal or analog. The charge pump 102, loop filter 103 and VCO 104 are regarded as analog building blocks while the feedback divider 105 and phase detector 101 are regarded as being either digital or analog building blocks. Unfortunately, the design of analog and (at least the analog portion of) mixed signal circuit design has traditionally involved manual, hand crafted techniques rather than automated techniques. Here, the manual, hand crafted approach to analog and mixed signal circuit design tends to slow down or otherwise complicate the circuit design flow process.

[0023] With semiconductor minimum feature sizes approaching 0.10 micron and below, highly integrated semiconductor chips are expected that will integrate significant amounts of both digital signal processing circuitry and analog signal processing circuitry onto the same semiconductor chip. Furthermore, with the growth of networking and handheld applications,

analog signal processing has observed a surge in interest (because communication and handheld circuits tend to depend more upon analog/mixed signal functionality than those used for desktop or raised floor computing system applications).

[0024] As such, analog and mixed signal circuit design techniques have received widespread attention recently because of the relative scarcity of analog circuit designers (in light of the increased demand for analog and mixed signal circuit designs themselves) combined with the manual, hand crafted approach to analog and mixed signal circuit design. Better said, the combination of manual design techniques, a small supply of analog designers and the increased demand for analog and mixed signal circuit designs has threatened the ability of analog/mixed signal circuit design teams to meet demand in a timely fashion. PLLs, being regarded as one of the more commonly used analog or mixed signal circuits, therefore tend to complicate or extend the development strategy of the many semiconductor chips because of the manual, hand crafted approach to PLL circuit design.

## **LIST OF FIGURES**

**[0025]** The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

**[0026]** Figure 1 shows an embodiment of a phase locked loop;

**[0027]** Figure 2Aa shows an embodiment of static phase error;

**[0028]** Figure 2B shows an illustration of stability;

**[0029]** Figure 2C shows an example of systematic jitter;

**[0030]** Figure 3A shows an embodiment of a netlist for an operational amplifier;

**[0031]** Figure 3B shows an embodiment of a transistor level design for an operational amplifier;

**[0032]** Figure 4A shows an example of a characterization of an op amp at the building block level of detail;

**[0033]** Figure 4B shows an embodiment of an op amp at the building block level of detail;

**[0034]** Figure 5A shows an embodiment of a methodology that may be used to develop a more detailed family of posynomial and monomial equations for a PLL;

**[0035]** Figure 5B shows an embodiment of a methodology that may be used to generate more detailed circuit topology information for a PLL;

**[0036]** Figure 6A shows an embodiment of a system level family of equations for a PLL;

**[0037]** Figure 6B shows an embodiment of the variables that are called out by the family of equations of Figure 6A;

**[0038]** Figure 6C shows an embodiment of a circuit topology for a PLL;

[0039] Figure 6D shows an embodiment of a netlist for the PLL topology of Figure 6c;

[0040] Figure 6E shows how a family of equations may expand if multiple operating points are described;

[0041] Figure 7A shows equations for each of a phase detector, a loop filter and a feedback divider;

[0042] Figure 7B shows circuit a topology for each of a phase detector, loop filter and feedback divider;

[0043] Figure 8A shows a first set of equations for a charge pump;

[0044] Figure 8B shows a second set of equations for a charge pump;

[0045] Figure 8C shows a circuit topology for a charge pump;

[0046] Figure 9A shows a first set of equations for a VCO;

[0047] Figure 9B shows a second set of equations for a VCO;

[0048] Figure 9C shows a circuit topology for a VCO;

[0049] Figure 10 shows a methodology by which an automated design tool may operate according to;

[0050] Figure 11 shows an embodiment of a methodology for creating a system level circuit topology;

[0051] Figure 12 shows an example of the effects of making a substitution into a posynomial or monomial equation;

[0052] Figure 13 shows an example of the effects of making a circuit topology substitution into a netlist;

[0053] Figure 14 shows an example of different thread lengths and thread paths for drilling down to a transistor level of detail;

[0054] Figure 15 shows an embodiment of a computing system.

## **DETAILED DESCRIPTION**

### **1.0 Introduction**

[0055] A solution to the problem of inefficient PLL signal circuit design is to automate the PLL circuit design process. Here, the ability to describe electronic circuitry at various levels of detail lends itself nicely to an automated circuit design process for a few reasons. Firstly, efficiency through automation is realized because a designer can adequately describe the PLL at a “high level” of detail. By allowing a designer to adequately describe the PLL at a high level of detail, the automated design process (rather than the designer) is left with the task of determining the more intricate details needed to prepare the PLL for manufacturing.

[0056] Here, the designer becomes more efficient because time that would have otherwise been devoted to determining the intricate details of the PLL design can be used, instead, to developing another high level description for another circuit design. Secondly, defining circuitry at various levels of detail makes the process flow of a software program that performs automated circuit design easier to define and construct. Better said, it is easier to develop a software program that is designed to automatically convert circuitry described at a first level of detail into circuitry described at a second level of detail (as opposed to an approach where various levels of detail are not appreciated). The following description discussed various techniques for an automated PLL circuit design process.

[0057] With respect to a PLL circuit design, completion of an automated circuit design process may be realized once a transistor level netlist has been formed. A transistor level netlist is a listing of the nodes (i.e., “nets”) in a circuit to be manufactured; and, for each node, a specification of each of the components that are connected to that node (e.g., a transistor gate, a transistor source, a transistor drain, a first electrode of a capacitor, a second electrode of a



capacitor, a first electrode of a resistor, a second electrode of a resistor, etc.). Netlists may also list the components themselves that are part of the circuit design (e.g., transistors, resistors, capacitors).

**[0058]** For each identified transistor, the netlist also specifies relevant transistor dimensions. For example, with respect to circuit designs targeted for a Complementary Metal-Oxide-Semiconductor (CMOS) field effect transistor manufacturing process, a transistor level netlist should define the gate length and gate width of each transistor. An example of a transistor level netlist 300 for an 8 transistor operational amplifier (“op-amp” or “op amp” or “opamp”) is observed in Figure 3a. The corresponding transistor level circuit design 350 of the op-amp to which the netlist 300 of Figure 3a pertains is observed in Figure 3b.

**[0059]** Note that a listing of each of the transistors M1 through M8 is provided in field 301 of the netlist 300; and, a listing of each of nodes 1 through 10 is observed in field 302 of the netlist. Note also that, for each of the transistors specified in field 301 of the netlist 300, the gate length and gate width have been specified. Furthermore, as seen in the exemplary netlist 300 of Figure 3a, note that other specifics are provided for such as specific resistance values and capacitance values (which may alternatively be described in terms of their dimensions), etc. It is important to point that the netlist 300 observed in Figure 3a (as well as those presented throughout the application) is an exemplary embodiment. As such, other netlists may be readily created by those of ordinary skill having different organization and/or format than that observed in Figure 3a.

**[0060]** Heretofore, it has been appreciated that geometric programming can be used to rapidly define specific transistor dimensions (such as gate width and gate length) for a particular transistor level circuit design (such as the op-amp observed in Figure 3b). For example, as

described in U.S. Patent No. 6,269,277B1 entitled “System and Method For Designing Integrated Circuits” and issued on July 31, 2001, specific transistor dimensions (as well as specific resistance and capacitance values) can be automatically determined for an operational amplifier based upon designer articulated functional characteristics of the operational amplifier. For example, if a user desires the operational amplifier to have an open loop gain of CdB, a unity gain bandwidth of DMHz, a slew rate of E V/nsec (to list just a few of the functional characteristics for an op amp), a software program may be used to mathematically relate the functional characteristics of the operational amplifier to specific transistor dimensions (and resistance and capacitance values) through “geometric programming”. As such, some degree of automated analog circuit design has been heretofore recognized.

[0061] However to the extent that geometric mathematics have already been identified as an underlying approach to automated analog circuit design; up to now, a “hierarchical” geometric approach has not been described or recognized. Here, a hierarchical geometric approach would allow specific transistor dimensions to be determined by “drilling down” through multiple levels of circuit description detail wherein each lower level adds more specific information about the circuit and wherein principles of geometric mathematics are preserved throughout each of the levels. Before describing such a system, a brief review of geometric programming will be provided so that the reader may understand relevant concepts relating to geometric programming.

## 2.0 Geometric Programming

[0062] Use of the term “geometric” in the present discussion refers to the use of equations that are expressed in monomial or posynomial form so that a geometric optimization problem can be constructed and/or solved in software. A monomial equation is an equation in the form of

$$cx_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n} \quad \text{EQN. 1}$$

where  $x_1, x_2, \dots, x_n$  are real, positive variables;  $c > 0$ ; and,  $\alpha_i$  is real. A posynomial equation is an equation in the form of

$$\sum_{k=1}^l c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \dots x_n^{\alpha_{nk}} \quad \text{EQN. 2}$$

where  $x_1, x_2, \dots, x_n$  are real, positive variables;  $c_k > 0$ ; and,  $\alpha_i$  is real. An example of an equation expressed in monomial form is  $2.3(x_1/x_2)^{1.5}$  (where  $c = 2.3$ ,  $\alpha_1 = 1.5$ ,  $\alpha_2 = -1.5$ ); and, an example of an equation expressed in posynomial form is  $0.7 + 3x_1/x_2^3 + x_2^{0.3}$  (where  $c_1 = 0.7$ ,  $\alpha_{11} = \alpha_{21} = \alpha_{31} = 0$ ;  $c_2 = 3$ ,  $\alpha_{12} = 1$ ,  $\alpha_{22} = 0$ ,  $\alpha_{32} = -3$ ;  $c_3 = 1$ ,  $\alpha_{13} = 0$ ,  $\alpha_{23} = 0.3$ ,  $\alpha_{33} = 0$ ).

[0063] Typically, a plurality of monomial and/or posynomial equations are collected or otherwise identified so that one of the equations can be “optimized” (e.g., maximized or minimized) so as to identify specific numeric values for the constituent variables associated with the equations used to construct and solve a geometric problem. For example, continuing with the example of the operational amplifier of Figures 3a and 3b, the power consumption, silicon surface area consumption, open loop gain, unity gain bandwidth, phase margin, and slew rate of the operational amplifier (as well as other functional characteristics of the operational amplifier) could each be described with a different monomial or posynomial equation.

[0064] As just one example of these, the open loop gain of the op amp could be expressed in monomial form as:

$$AO = KIBIAS(W_2W_6L_7W_8L_5W_8/L_2L_6W_7L_8W_5L_8)^{0.5} \quad \text{EQN. 3}$$

where: 1) K is a constant obtainable by those or ordinary skill; 2) IBIAS is the current drawn by the current source 310 of the op amp of Figure 3b; 3) W is the gate width of the transistor that the applicable subscript identifies; and; 4) L is the gate length of the transistor that the applicable subscript identifies. Here, other monomial or posynomial equations that each express a specific op amp characteristic (such as those listed in the preceding paragraph (as well as others)) in terms of variables that correspond to transistor gate dimensions (or other parameters that can be converted into specific transistor dimensions as described in more detail below) can be “collected” so that the collection of equations represents a family of equations that, together, describe (in monomial and/or posynomial form) the operational amplifier as a whole.

[0065] Once a group of appropriate monomial equations and/or posynomial equations are collected, specific numeric values (i.e., “constraints”) may be specified for the particular functional characteristic that each of the equations describe. In a common approach, one of the equations is not provided a specific numeric constraint so that it can be optimized (in respect of the numeric constraints applied to the other equations) through a geometric solving process. That is, in this case, the geometric problem may be constructed according to the following definition:

*Minimize* : monomial or posynomial equation\_z

*Subject to:*

constrained form of monomial equation\_1

constrained form of monomial equation\_2

.

.

.

constrained form of monomial equation\_X

AND

constrained form of posynomial equation\_1

constrained form of posynomial equation\_2

constrained form of posynomial equation\_Y                      EQN. 4

[0066]     Here, the “family” of posynomial and monomial equations used to describe the circuit (such as the op amp referred to above) include equation z (which may be from the group of monomial equations in the family or from the group of posynomial equations in the family). As such, there are  $X+Y+1$  total equations in the family of equations; one of which is to be optimized, the remainder of which are constrained.

[0067]     As an example, if specific numeric “target” values (or ranges) are articulated for each of the op amp characteristics other than silicon surface area consumption (e.g., power    35mW, open loop gain > 106; unity gain bandwidth    100MHz; phase margin > 60°; slew rate    2.5 v/nsec), a geometric optimization problem can be constructed and solved wherein the solution corresponds to the minimum silicon surface consumption for the particular numeric constraints that were articulated. Better said, in the geometric problem model provided just above, the equation for silicon surface area consumption corresponds to “equation\_z”; and, the remaining equations that describe the functional characteristics of the op amp correspond to the group of  $X+Y$  equations within the family that have been numerically constrained.

[0068]     As such, if the equation for silicon surface area consumption is expressed in terms of specific transistor dimensions, the solution to the geometric problem (once the geometric problem is solved) will specify specific transistor dimensions that not only correspond to the minimum area of the op amp that may be achieved but also correspond to an op amp having the

specific numeric constraints that were articulated. Before continuing, note that above cited geometric problem is just one type of geometric problem. For example, other geometric problems may choose to allow one or more equations (that take part in the problem solving process) to be unconstrained and neither maximized nor minimized. Generally, geometric problems are optimization problems that employ posynomial and/or monomial equations (that may be expressed in convex form or otherwise as explained in more detail below). The reader is referred to S. Boyd and L. Vandenberghe, "Introduction to Convex Optimization With Engineering Applications", Course Notes, 1999, <http://www.leland.stanford.edu/class/ee364> for more details regarding the various forms by which an optimization problem may be expressed in geometric form.

**[0069]** Regardless as to the specific form in which a geometric optimization problem is stated, note that the manner in which that form is obtained (e.g., the techniques or strategies used to develop a family of specific monomial and/or posynomial expressions) may be widely varied. For example, note that the IBIAS term that appeared above in the (monomial) expression for open loop gain (EQN. 3 may be further expressed in monomial or posynomial form in terms of the dimensions of its constituent transistors (which have been obviated from Figure 3b for simplicity).

**[0070]** As such, such an expression for IBIAS could be: 1) directly substituted into EQN. 3 (so as to form an expression for open loop gain "completely" in terms of a constant and transistor dimensions); 2) "kept apart" as a separate equation that is included with the family of equations that take part in the geometric problem solving process (e.g., is one of the X+Y equations illustrated in EQN. 4); or, 3) "kept apart" as a separate equation that is not included with the

family of equations that take part in the geometric problem solving process (e.g., is not one of the  $X+Y$  equations illustrated in EQN. 4).

[0071] In the former (first) case, the “variables” of the geometric optimization process (at least with respect to open loop gain) are limited to transistor dimensions as expressed in a single equation. In the middle (second) case, the “variables” of the geometric optimization process (at least with respect to open loop gain) are limited to transistor dimensions as expressed via a pair of equations (e.g., one for open loop gain and one for bias current). In the later (third) case, the “variables” of the geometric optimization process (at least with respect to open loop gain) include not only transistor dimensions but also the bias current IBIAS of the op amp current source.

[0072] In the later case, once an optimal value for IBIAS is obtained (e.g., as would naturally “fall out” as a result of a solution to a geometric problem where IBIAS was one of its variables), a separate following sequence (geometric or otherwise) could be used to determine specific transistor dimensions for those transistors used to construct the op amp current source 310. Thus, a complete description for the op amp could be obtained even if the posynomial and monomial equations that are used to construct a geometric problem are not expressed “solely” in terms of transistor dimensions. This feature may be taken advantage of, for example, in order to express performance characteristics in more traditional terms ( e.g., where, in the case of op amps for example, important functional characteristics tend to be explained in terms of op amp bias current rather than the dimensions of the transistors that form the current source that provides the bias current).

[0073] The fact that the open loop gain of an op amp could be accounted for in a geometric programming environment by at least three different approaches has been mentioned above to

bring to light some relevant properties regarding the application of geometric programming techniques to analog circuit design automation. Firstly, the specific organization as to which functional characteristics are expressed as a separate equation and/or which variables are chosen to take part in each equation (i.e., how each posynomial and monomial equation is particularly expressed) and/or which equations take part in a geometric problem solving sequence (and which equations simply “make use of” an optimization result) may vary from embodiment to embodiment.

[0074] As such, for any analog or mixed signal circuit design, an automated design tool may employ any of a vast number of different, mathematical “paths” or “threads” in order to achieve a specific end result (e.g., transistor gate dimensions for the transistors within the circuit). Here, the simple case of op amp open loop gain has illustrated three unique mathematical paths that could be used. As such, where specific mathematical equations (or specific sequence(s) of specific mathematical equations) have been presented in the present application (e.g., as an example of a process flow for an automated design tool), the claims that follow should not be automatically construed as being limited solely to these equations (or to the specific sequence(s) according to which they flow).

[0075] Secondly, a second property that was touched upon by the former (first) approach (by which open loop gain was expressed completely in terms of transistor dimensions by substituting an expression for IBIAS into EQN. 3) is that posynomial or monomial equations may be “substituted” into other posynomial or monomial equations such that the resultant equation preserves a posynomial or monomial form. For example, if the equation for IBIAS is expressed in monomial form, its substitution into EQN. 3 would cause the resultant expression for open loop gain to also be in monomial form (since EQN. 3 is expressed in monomial form). Similarly,



the substitution of a monomial expression into a variable of a posynomial expression allows the posynomial expression to remain in posynomial form; and, the substitution of a posynomial expression into a monomial expression causes the monomial expression to expand into a posynomial expression.

[0076] Before continuing to the next section, it is important to point out that (once a family of equations in monomial and/or posynomial form have been collected for a particular analog circuit) those of ordinary skill will be able to readily construct an appropriate geometric problem in software; and, furthermore create or use known methods to actually solve the geometric problem in software. For example, according to one type of approach referred to as the “convex” approach, logarithms of monomial and posynomial expressions (whose variables are expressed as exponentials) are used in order to convert the geometric optimization problem into convex form.

[0077] A convex optimization problem effectively optimizes a logarithmic function in light of a plurality of other, constrained logarithmic functions. Here, optimization problems expressed in convex form are numerically easier to compute for a digital computing system (e.g., a computer) that is tailored to solve the optimization problem (e.g., through the use of a software program). Accordingly, the term “geometric problem”, “geometric optimization problem” and the like is meant to include geometric problems that have been formulated in convex form. Likewise, the term monomial is meant to include monomial equations that are in a format suitable for a convex optimization problem; and, the term posynomial is meant to include posynomial equations that are in a format suitable for a convex optimization problem. Typical software approaches used to solve convex optimization problems include interior-point primal barrier methods, interior-point primal-dual methods, and infeasible primal-dual methods among

others. For these or other details related to geometric programming, the reader is again referred to <http://www.leland.stanford.edu/class/ee364/>.

### 3.0 Hierarchical Approach to Geometric Programming Based Automated PLL Circuit Design

#### 3.1 Basic Analog Building Block Level of Detail

**[0078]** Up to now, as alluded to above, known geometric problem solving techniques for automated analog circuit design have been limited to the transistor level. As such, geometric problem solving techniques for automated analog circuit design have been limited in application to defining transistor dimensions only for isolated, basic analog building blocks. A basic analog building block is an analog circuit (or mixed signal circuit) whose “next”, lower level of detail (i.e., more specific level of detail) is a transistor level of detail when the analog circuit (or mixed signal circuit) is not represented at a transistor level of detail.

**[0079]** An embodiment of a basic, analog building block level of detail (or simply, “basic building block level of detail”) for an operational amplifier is observed in Figures 4a and 4b. Figure 4a shows an embodiment of a functional description articulated at the basic building block level of detail; and, Figure 4b shows a circuit topology depiction for an opamp at the basic building block level of detail. Here, the opamp descriptions of Figures 3a and 3b may be compared with the opamp description of Figures 4a and 4b so as to form an understanding as to what is meant to by the term “basic building block” as described just above.

**[0080]** The operational amplifier depiction 450 of Figure 4b is drawn as operational amplifiers are commonly drawn in a schematic drawing of a circuit that employs an operational amplifier. The characteristic description 400 of the operational amplifier 450 of Figure 4b is observed in Figure 4a. At the level of detail observed in Figure 4b, with the exception of silicon

surface area consumption, the description 400 of the operational amplifier 450 observed in Figure 4a is limited to a listing of “functional” characteristics. Functional characteristics are electrical in nature rather than physical in nature. Thus, as seen in the example of Figure 4a, the units of the characteristics listed therein are milliwatts (mW), decibels (dB), megahertz (MHz), degrees of phase, and Volts per nanosecond (V/nsec), etc.

[0081] Here, because the level of detail of the circuit topology 450 observed in Fig. 4b is devoid of transistor level information, the corresponding level of detail level to which the characteristics of the operational amplifier can be defined is correspondingly limited to levels of detail that are higher than the transistor level (such as the functional characteristics referred to above and observed in Figure 4a). The next, lower level of detail to which the operational amplifier of Figures 4a and 4b can be described is the transistor level of detail observed in Figures 3a and 3b.

[0082] As discussed in the preceding section, the transistor level of detail includes transistor level information such as a circuit topology 350 that shows the individual transistors M1 through M8 used to construct the operational amplifier; and/or, a transistor level netlist 300 that not only describes the transistor level circuit topology 350 of Figure 3b, but also includes specific dimensions for the transistors themselves. As such, whereas the level of detail associated with the description 400 of Figure 4a describes the op amp largely in terms of its functional characteristics rather than its physical characteristics, note that the transistor level of detail describes the op amp largely in terms of its physical descriptions (e.g., specific nodes, their connections to particular transistor level components (such as resistors, capacitors and transistors) and the dimensions of these components) rather than its functional characteristics.

[0083] Because the next, lower level of detail of the description provided in Figures 4a and 4b is the transistor level (as observed in Figures 3a and 3b), the level of detail observed in Figures 4a and 4b correspond to a “basic building block”. That is, because the next, higher level of detail above the transistor level of detail of Figures 3a and 3 is the level of detail observed in Figures 4a and 4b, the level of detail observed in Figures 4a and 4b is a “building block” level of detail. Here, referring back to Figure 1, note that a PLL may be viewed as being constructed of five basic building blocks: 1) a phase detector 101; 2) a charge pump 102; 3) a loop filter 103; 4) a VCO 104; and, 5) a feedback divider 105.

[0084] Thus, to summarize, as is presently recognized in the prior art, geometric programming techniques have been used to automatically convert a description of an analog circuit at the basic building block level into a description at the transistor level of detail. Although this has commercial application, analog circuitry is often used, implemented and/or otherwise viewed at levels that are “higher” than the mere building block level. The PLL is an example wherein the PLL 100 is viewed as overall working system that is made up of its constituent building blocks 101 through 105.

[0085] PLLs may be described according to their output signal frequency range, static phase error, stability, jitter, etc. Note that, consistent with the concept of an analog system, these functional characteristics tend to describe properties of the overall PLL system that is formed by the coupling of various basic building blocks 101-105 as depicted in Figure 1 (rather than describing the functional properties of a single building block by itself).

#### 4.0 Methodologies For Automatically Developing A Transistor Level Information For A PLL From The System Level

[0086] Figure 5a shows a unit of operational flow 500 for a design tool that automatically develops a design for a PLL. According to the basic unit of operational flow observed in Figure 5a, a variable within a posynomial or monomial expression for a characteristic of a PLL (e.g., static phase error, cross over frequency, phase margin, jitter, etc.) is recognized 501 as having a dependency on a lower level expression. Here, the dependency may be recognized (for example) because the variable is not expressed in terms of transistor level details or is otherwise identified as having information described at a level that is higher than the transistor level.

[0087] The term “dependency” is used to invoke the notion that, in order for the variable to be represented at the transistor level, more detailed information needs to be gathered (e.g., by “drilling down” through multiple levels of detail). Once a system level variable of a PLL has been identified 501, a lower level monomial or posynomial expression for that variable is substituted 502 into the system level expression (so as to replace the “substituted for” variable within the system level expression). Substitutions may then be continually made until the original system level variable is expressed in transistor level terms.

[0088] Figure 5b shows another unit of operational flow 550 for a design tool that automatically develops a design for a PLL. A distinction between the methodology 500 of Figure 5a and the methodology 550 of Figure 5b is that the methodology 500 of Figure 5a relates to the manipulation of posynomial and/or monomial equations; whereas, the methodology 550 of Figure 5b relates to the manipulation of a netlist (or other circuit topology description) that describes the specific components and/or nodes of a circuit design. According to the basic unit of operational flow 550 observed in Figure 5b, a component within the circuit topology of an analog or mixed signal system is recognized 551 as having a dependency on a lower level circuit component (e.g., a basic building block such as the VCO). Here, the dependency may be

recognized (for example) because the circuit component is not expressed in terms of transistor level details or is otherwise identified as having information described at a level that is higher than the transistor level.

[0089] The term “dependency”, again, is used in the above to invoke the notion that, in order for the component to be represented at the transistor level, more detailed information needs to be gathered (e.g., by “drilling down” through multiple levels of detail). Once a component of a PLL system has been identified (e.g., the VCO) 551, a lower level description of that circuit component is substituted 552 into the system level circuit topology (so as to replace the “substituted for” component within the system level topology). Substitutions may then be continually made until the original system level component is expressed in transistor level terms.

[0090] The methodologies 500, 550 of Figures 5a and 5b may be executed in series or in parallel, alone or in combination in order to help develop multiple “paths” or “threads” of circuit description detail so that, eventually, a PLL is thoroughly described at a transistor level of detail. As described further detail below, the various threads may “branch out” from a common thread section and/or exhibit longer thread lengths than other thread lengths (signifying a greater number of substitutions having to be made to reach the transistor level of detail). Note also that, according to various possible approaches, either methodology 501 alone (dependent variable recognition) or methodology 551 alone (dependent component recognition) may be used to trigger the execution of both methodologies 502 (equation substitution) and 552 (component substitution). This follows naturally when one views circuit topology and functional/physical description as part of a complete description (noting that at the transistor level topology and description may become merged as discussed with respect to Figure 3a).

## 5.0 Embodiment of a System Level Description For A PLL

### 5.1 Overview

**[0091]** Figures 6a through 6e relate to an exemplary embodiment of a system level PLL description that can be “drilled down” to a transistor level of detail by employing the methodologies 500, 550 of Figures 5a and 5b. Specifically, as explained in more detail below, the system level description for the PLL may be made to include: 1) a family 600a of equations expressed in monomial or posynomial form (e.g., as seen in Figure 6a); and 2) a netlist 600d (e.g., as observed in Figure 6d), or other circuit description that describes the topology 600c of a PLL (e.g., as seen in Figure 6c). Note that the level of detail associated with the descriptions referred to just above corresponds to a system level of detail because the PLL is described (both in the equation family 600a and in the topology 600c) as a collection of basic building blocks (i.e., phase detector, charge pump, loop filter, VCO and frequency divider).

**[0092]** The effect(s) that the aforementioned drilling down activity will have, in terms of enabling the automatic incorporation of more detailed information into the description of the PLL, will become more apparent in the discussion that follows. Figure 6a shows a representation of a family 600a of posynomial and monomial equations 621 through 638 that describe a PLL at the system level. Figure 6b shows the collection of variables and user defined values that are referred to by the family of equations 600a of Figure 6a. Figure 6c shows a system level circuit topology 600c for the PLL that may be described by a netlist (or other type of circuit description that may be used to describe a circuit topology). Figure 6d shows an embodiment of a netlist that may be used to describe the PLL topology of Figure 6c. Figure 6e shows an example of how the family of equations 600 of Figure 6a may be expanded so as to account for different operating points of the PLL.

[0093] Referring to Figure 6c, the building blocks that make up the PLL 600c includes a phase detector (PFD) 601c, a charge pump (CP) 602c, a loop filter (LF) 603c, a voltage controlled oscillator (VCO) 604c and a feedback divider (DIV) 605c. Initially, an automated design tool may be tailored to configure (e.g., automatically or with the help of a designer) the system level circuit topology 600c of Figure 6c (e.g., with a netlist such as that observed in Figure 6d). Referring to either of Figures 6a, 6b, or 6c, note that (with the exception of the LF.C1, LF.C2 and LF.R terms) transistor level information is devoid from the descriptions provided therein.

[0094] As will be made apparent in subsequent sections of this application, as a result of the design tool automatically “drilling down” through multiple levels of detail, multiple threads of circuit level detail are developed. Here, each thread should extend from the high level, system representation of the PLL (e.g., as embodied in the equation set 600a Figure 6a and the netlist 600d of Figure 6d) toward (finally) a transistor level description. Thus, in the sections following the present section, each of the PLL’s building blocks 601c through 605c will be examined in greater detail. Here, an embodiment of a transistor level description (or at least a portion thereof), as implemented with additional transistor level posynomial/monomial equations and a circuit topology (from which a netlist can be readily developed) will be provided for each building block 601c through 605c. As such, the reader should be able to gain an appreciation as to how a software design tool can automatically develop a transistor level description of a PLL.

[0095] Thus, for each building block 601c through 605c, additional monomial and/or posynomial equations will be provided; wherein, these additional equations may be substituted (e.g., in accordance with the methodology 500a of Figure 5a) into their corresponding variable within the family of equations 600a of Figure 6a (in order to expand upon the system level



equation set 600a). In some cases, as examples, the result of these aforementioned substitutions will be shown so that the reader can gain an understanding into how a system level set of equations 600a for a PLL may be made to expand into greater detail as the PLL is described at (or at least closer to) a transistor level of detail. Similarly, a transistor level netlist for the loop filter 603c, a transistor level netlist for the VCO 604c and a transistor level netlist for the charge pump 602c will be substituted into the system level netlist 600d of Figure 6d (e.g., in accordance with the methodology 550 of Figure 5b) so that the reader can gain an understanding into how a system level circuit topology description may be made to expand into greater detail as the PLL is described at a transistor level of detail. As such, the description that follows is written with an eye toward the methodologies 500, 550 observed in Figures 5a and 5b.

## 5.2 Family of System Level Monomial/Posynomial Equations

[0096] With an understanding that a system level description may be embodied with a PLL system level netlist (e.g., as depicted in Figure 6d) that describes a PLL system topology (e.g., as depicted in Figure 6c) in terms of its constituent building blocks 601c through 605c, an examination into an embodiment of family of monomial and/or posynomial equations 600a (as depicted in Figure 6a) is presently appropriate. Referring then to Figure 6a, note that the family of equations are organized into: 1) an equation 621 that describes the semiconductor surface area consumption of the PLL; 2) an equation 622 that describes the power consumption of the PLL; 3) a sub-family of equations 623, 624, 625, 626 that describe the static phase error of the PLL; 4) a sub family of equations 627, 628, 629, 630 that describe the output frequency of the PLL; 5) a sub family of equations 631, 632, 633, 634 that describe the stability of the PLL; 6) a pair of equations 635, 636 that describe the peak jitter of the PLL; and, 7) a pair of equations 637, 638

that describe the sensitivity of the PLL to fluctuations in the supply voltage VDD (which may be referred to as power supply rejection). A discussion of each of these follows below.

#### 5.2.1 area and power consumption.

[0097] Consistent with the PLL being described at a system level of detail, the PLL's semiconductor surface area consumption is described in equation 621 in terms of constituent basic building blocks 601c through 605c. That is, the area of the PLL is set equal to the summation of a variable that describes the area of the phase detector (PFD.AREA), a variable that describes the area of its charge pump (CP.AREA), etc. Note that equation 621 is in posynomial form, assuming the area of all constituent blocks are also in posynomial form. Similarly, the PLL's power consumption is also described in equation 622 in terms of its constituent basic building blocks 601c through 605c. That is, the power consumption of the PLL is set equal to the summation of a variable that describes the power consumption of the phase detector (PFD.AREA), a variable that describes the power consumption of the charge pump (CP.AREA), etc. Note that equation 622 is also in posynomial form.

#### 5.2.2 static phase error

[0098] A sub family of equations 623, 624, 625, 626 are used to describe the static phase error of the PLL (recall that static phase error was discussed in the background at Section 2.1). Note that equations 623 and 625 are posynomial while equations 624 and 626 are monomial. Equation 623 defines the static phase error for the PLL in terms of various phase detector and charge pump related variables (e.g., noting the PFD. and CP. based variables) as observed in terms 623b, 623c, and 623d; and a variable PLL.DELTA\_T\_STD\_DEV 623a that is further

described by equations 624 and 625. Note that equation 624 merely takes the square root of the quantity defined by equation 625. Examining equation 625 in further detail it is clear that equation 625 also uses phase detector and charge pump related variables. As such, the static phase error for the PLL is described in terms of various characteristics associated with the PLL's phase detector and charge pump.

**[0099]** The static phase error of a PLL is typically viewed as having random components and systematic components. Here, the systematic components are expressed through terms 623b, 623c and 623d of equation 623 while the random components are expressed through equation 625 (whose summation undergoes a square root operation via equation 624 in order to equalize the units of the random terms with those of the systematic terms). Referring to the systematic terms first, term 623b (PFD.DELTA\_TERROR) effectively describes a fixed time difference between the pair of phase detector outputs ("up" and "down"). Better said, the phase detector (e.g., as a result of different path lengths within the phase detector) may fixedly signify a positive error term larger than (or smaller than) the expected error term. The PFD.DELTA\_TERROR term 623b corresponds to the speed difference (e.g., in seconds).

**[00100]** Term 623c effectively describes a difference, per error term, between the charge that is added to the loop filter and the charge is removed from the loop filter. The charge pump 602c, according to one embodiment, is designed with a current pushing circuit section that adds charge to the loop filter and a current pulling circuit section that extracts charge from the loop filter. Furthermore, according to a further embodiment, for each error term, the phase detector is designed to turn "on" both of these circuit sections (i.e., both the "up" and the "down" outputs are activated), for a fixed period of time PFD.TRESET regardless of the error term magnitude or polarity.

[00101] Here, a fixed difference between the amount of charge that is added and the amount of charge that is removed per error term over the course of the PFD.TRESET time will generate static phase error because, in a sense, an error term will be created from the perspective of the VCO when the phase detector issues an error term of zero magnitude; and, likewise, an error term will not be created from the perspective of the VCO when the phase detector issues an error term that perfectly offsets the difference in charge described above. Here, CP.DELTA\_IP corresponds to the difference in current generated by the current pushing and current pulling charge pump circuit sections described just above; and, CP.IP corresponds to the current that is pushed by the current pushing section; or, equivalently, the current that is pulled by the current pulling section.

[00102] Term 623d effectively describes charge that is added (or removed) from the loop filter, by the charge pump, for reasons other than the phase detector's simultaneous activation of both the "up" and "down" inputs over the PFD.TRESET time period. For example, as described in more detail below with respect to the transistor level description of the charge pump, the charge pump may be designed to include a voltage follower circuit between the current pushing and current pulling sections that undesireably adds charge to the loop filter (e.g., due to an offset voltage in an operational amplifier that is used to implement the voltage follower). Here, the term CP.DELTA\_QSTAT represents the amount of this charge that is added per unit of pushing current or pulling current (CP.IP).

[00103] The causes of static phase error described just above may exist not only in a fixed, systematic fashion - but also in a random fashion (e.g., with wafer to wafer variation). Equation 625 helps summarize these random contributions to static phase error. As such, variables such as CP.IP\_VARIANCE and CP.VARIANCE\_QSTAT (which are described in more detail below

with respect to the transistor level description of the charge pump) are employed. Note that, consistent with principles of statistics, the variances of independent, random occurrences may be directly added. Equation 625 corresponds to the summation of the variances of the causes described just above. Here, as variance corresponds to the square of the standard deviation of a random occurrence, equation 624 takes the root of the variance so that that the units are the same for each of the terms 623a through 623d in equation 623 (e.g., seconds) rather than disparate (e.g., seconds<sup>2</sup> for random terms and seconds for systematic terms). Note that the user defined constant  $\sigma$  in term 623a can be used to define the peak static phase error in terms of a number of standard deviations of the PLL.DELTA\_T\_STD\_DEV

**[00104]** Finally, equation 626 allows the circuit designer to specify an upper limit for the static phase error of the PLL that is to be designed. Here, by specifying that the static phase error should be less than or equal to a certain amount of time, a specific posynomial constraint will be applied to equation 626. As such, when equation 626 participates in the geometric problem solving sequence, the transistor dimensions that will be defined as a result of the geometric problem solving process will automatically correspond to a PLL having a static phase error that falls within the range originally specified by the user via equation 626.

### 5.2.3 PLL output frequency

**[00105]** Equation 627 specifies the PLL output frequency as being equal to the output frequency of the VCO. Equation 630 also specifies the PLL output frequency in terms of the feedback division (DIV.M) performed by the feedback divider 605c and the frequency of the input reference signal PLL.INPUT\_REF\_SIGNAL\_FREQ. Both of these follow directly from the topology 600c of the PLL as well as basic principles of PLL operational theory. Equations

628 and 629 allow the designer to specify a minimum and maximum operating frequency for the PLL. For example, if the designer's application can tolerate some slack in the specific output PLL output frequency (e.g., 1.000 GHz +/- .002 GHz) the user may specify the frequencies at either end of the allowable frequency range (e.g., .998 GHz at the low end and 1.002 GHz at the high end).

[00106] These, in turn, correspond to two different operating points that the PLL may be designed to work according to. Figure 6e illustrates the effect that multiple operating points may have on the family of monomial and posynomial equations. According to Figure 6e, as a result of a pair of operating points being specified by the designer (e.g., min PLL output frequency and max PLL output frequency), the family of monomial and posynomial equations 600a of Figure 6a may be expanded (e.g., doubled) such that the overall family of equations used for geometric problem solving 600e has a first set 600a1 that corresponds to the constraints that apply at the first operating point (e.g., the maximum allowable frequency range); and a second set 600a2 that corresponds to the constraints that apply at the first operating point (e.g., the minimum allowable frequency range).

[00107] Here, note that various system level characteristics may change in light of the different operating point whereas others may not change. For example, the power consumption of the PLL should be greater at the maximum PLL output frequency than at the minimum PLL operating frequency. As such, a different numeric quantity should appear for PLL.PWR in family set 600a1 as compared to family set 600a2. Here, if the designer were to specify a maximum allowable power consumption (e.g., with a PLL.PWR\_USER\_SPEC input which is not indicated in Figure 6a for simplicity), the geometric problem solving process would provide

a solution (if possible) that kept the power consumption within the specified range for each operating point.

**[00108]** Note that a nominal PLL output frequency specification (e.g., 1.000 GHz) could also be added (which would add a third family set of equations 600a3). Generally, users may define specifications for any of the variables in the family of equations. Some of these may be used to develop the PLL in light of specific environmental conditions; which, effectively correspond to different operating points as well (e.g., min, nominal and maximum supply voltage VDD; min, nominal and maximum temperature, etc.). So doing will add more constraints to the optimization process (which are represented by each additional family set). Some variables may not vary across different operating points (e.g., area); and, as such do not need to be replicated across each family set for each operating point.

#### 5.2.4 stability

**[00109]** The stability of the PLL is accounted for with equations 631 through 634. Note that equation 631, 633 and 634 are monomial while equation 632 is posynomial. Equation 631 describes the cross over frequency while equation 632 describes the phase margin. Recall that both cross over frequency and phase margin were discussed in the Background in Section 2.2. Here, the cross over frequency is expressed as the frequency where the magnitude of the transfer function through the loop is equal to unity. As the transfer function of the loop is the product of the transfer function for each of the phase lock loop's basic building blocks, note that equation 631 resembles the product of the "gain" for each of the phase detector (PFD.GAIN), charge pump (CP.IP), loop filter (LF.R) and VCO (VCO.GAIN) normalized by the "gain" of the feedback divider (DIV.M).

[00110] Equation 632 provides an expression for the phase margin. Here, as the phase of the PLL channel is mostly determined by the loop filter and the VCO, equation 632 is expressed in terms of the loop filter components LF.R, LF.C2 and a performance characteristic of the VCO that is related to the phase delay through the VCO (VCO.TAU3). Furthermore, as the phase margin corresponds to how far above -180 degrees the phase of the PLL channel is at the cross over frequency, note that the PLL cross over frequency PLL.CROSS\_OVER\_FREQ also appears in equation 632. Note that the embodiment of Figure 6a allows the user to specify both the cross over frequency and phase margin (via equations 633 and 634). Specifying these parameters will impose numeric constraints on equations 631 and 632; which, in turn, will result in a family of equations that will cause the geometric problem solving process to provide transistor level details for a PLL having the stability characteristics specified by the designer.

#### 5.2.5 peak jitter

[00111] The peak jitter of the PLL is accounted for with equations 635 and 636. Note that equation 635 is posynomial. Jitter was first discussed in the Background in Section 2.3. Peak jitter is, basically, an approximation of the maximum extent that jitter that will be apparent in the PLL output signal. Here, as jitter corresponds to the time variation of the VCO output signal frequency, the jitter will be apparent in the time domain, for example, by variation in the position of the PLL output signal's rising edges and/or falling edges. The maximum approximate extent of the variation of these edges corresponds to one perspective of peak jitter.

[00112] Peak jitter, like static phase error, may have random and systematic components. Here, term 635a expresses the random component of the peak jitter and term 635b expresses the systematic component of the peak jitter. The random component 635a effectively describes the



effect of thermal fluctuations in terms related to the gain of each of the various building blocks of the PLL; and, a special characteristic of phase noise in electrical oscillators that is expressed as the VCO characteristic variable VCO.KAPPA. The systematic term 635b describes the peak jitter that results from a pair of jitter noise sources that originate from the activity of the phase detector and charge pump.

[00113] As such, note that the systematic term 635b refers to phase detector related variables PFD.TRESET, PLL.INPUT\_REF\_SIGNAL\_FREQ; and, the charge pump related variable CP.DELTA\_IP. These jitter noise sources are effectively combined and then modified by the remainder of the transfer function of the PLL. As such, note that terms related to the “gain” of the loop filter, VCO and feedback divider (i.e., LF.R; VCO.GAIN; DIV.M) appear in term 635b. The transfer function of the PLL is further approximated with term 635b1. Here, constants  $\alpha$  and B are set to values that make the approximation more accurate. In an embodiment  $\alpha = 0.84$  and  $B = -1.95$ . The user can also specify an allowable range for peak jitter as observed with respect to equation 636.

#### 5.2.6 power supply rejection

[00114] Power supply rejection relates to the sensitivity of the PLL output signal frequency to fluctuations in the supply voltage VDD. Here, as the VCO is a voltage sensitive device and as the VCO provides the PLL output signal, this characteristic of the PLL (PLL.VDD\_DC), like the PLL output signal frequency in equation 627, is expressed in terms of the VCO itself (via VCO variable VCO.VDD\_DC). The user can also specify an allowable range for this PLL characteristic.

### 5.3. Variables Dependent of Lower Level Expressions

[00115] Figure 6b shows each of the variables that are called out by the family of equations 600a of Figure 6a. Figure 6b organizes these variables in terms of the specific building block to which they relate; or, alternatively, as a user defined variable (if the variable is to be defined by the user according to the embodiment of Figure 6a). As such, each of the variables associated with the phase detector 601b, charge pump 602b, loop filter 603b, VCO 604b and feedback divider 605b are represented in Figure 6b. Furthermore, each of the user defined variables are also listed 606b. Here, in accordance with methodology 500 of Figure 5a, the non user defined variables 601b through 605b would be recognized as having a dependency on a lower level expression (perhaps with the exception of the loop filter components LF.R, LF.C1 and LF.C2 because these may be viewed, according to some embodiments, as already being at the transistor level of detail).

## 6.0 Embodiments of PLL Building Block Profiles

### 6.1 Overview

[00116] The previous section related to an embodiment for describing a PLL at the system level in terms of its constituent building blocks. The present section describes embodiments for each the various building blocks (e.g., the phase detector, charge pump, loop filter, VCO and feedback divider) of which the PLL system is comprised. Each of these descriptions will provide more detailed monomial/posynomial expressions for each of the variables that the higher level PLL description discussed in the previous section depends upon. Furthermore, a circuit topology (from which a netlist could be readily derived) for each of the building blocks is also provided. As such, as described in a following section, the information presented in the present section

(Section 6.0) may be substituted into the information provided in the previous section (Section 5.0) in order to develop a description of a PLL toward a greater level of detail.

## 6.2 Phase Detector, Loop Filter and Feedback Divider

[00117] Figure 7a shows a monomial/posynomial equation family for the phase detector 701a, loop filter 703a and feedback divider 705a that further describe the variables that the system level PLL description discussed in Section 6.0 depends upon; and, Figure 7b shows a more detailed circuit topology embodiment 701b, 703b, 705b for each of these same building blocks. According to an embodiment, referring to Figure 7a, a “fixed” circuit design is used for both the phase detector 701a and the feedback divider 705a. As such, the properties of these building blocks are expressed as “constants” (i.e., do not vary) rather than variables. This simply means that these building blocks are not “optimized” by the geometric problem solving process. However, their properties (as represented by constants Const\_1 through Const\_9) take part in the geometric solving process.

[00118] That is, as constants Const\_1 through Const\_9 are numeric quantities that describe the corresponding characteristic to which they are labeled (e.g., Const\_1 for the phase detector power, Const\_2 for the phase detector area, etc.), their substitution into the PLL equation family set (e.g., as depicted in Figure 6a) will effectively add additional numeric constraints to the geometric optimization problem. Better said, the geometric problem solving process will attempt to optimize the charge pump components, loop filter components and VCO components in light of the numeric constraints (Const\_1 through Const\_9) that are imposed by use of a pre-configured, “fixed” phase detector and frequency divider design. This approach may be used if

it is believed that the charge pump and frequency divider (perhaps because they make use of digital circuitry) do not add much to the overall optimization issue.

[00119] That is, where multiple operating points and/or user specified design constraints will prove to mostly alter the charge pump, loop filter and VCO anyways. In such instances, the phase detector and frequency divider may as well be fixed to reduce computation complexity. As such, according to such an approach, a transistor level netlist (e.g., in an HSPICE format) is included with the design software. Here, for each potential target manufacturing process (e.g., IBM 0.13 micron, IBM 0.18 micron, TI 0.13 micron, TI 0.18 micron, etc.), the pre-configured phase detector and feedback divider may be simulated beforehand so that the set of constants Const\_1 through Const\_9 can be generated (e.g., for each design point specified by the designer). Note that each of the expressions associated with the phase detector 701a and feedback divider 705a are in monomial form. In alternative embodiments, the phase detector and/or feedback divider may be optimized along with the other building blocks. In this case, expressions written in monomial and/or posynomial having variables (as opposed to constants) should be used.

[00120] The expressions for the loop filter 703a describe the area of the loop filter in terms of the area of each of its components. Here, the area of each component is related to its electrical value (e.g., capacitance for capacitors C1 and C2 and resistance for resistor R) through a constant  $\beta$  for capacitors C1, C2 and another constant  $\gamma$  for resistance R. The specific capacitance and resistance values LF.C1, LF.C2 and LF.R are variables that are explicitly solved for by the geometric solving process (according to this embodiment) and therefore represent a lowest level of detail.

[00121] Referring to the circuit topologies 701b, 703b and 705b of Figure 5b, those of ordinary skill may readily develop a transistor level netlist for each. Here, according to the

embodiment discussed above wherein the phase detector and feedback divider are fixed, a netlist (e.g., in HSPICE format) for each of these building blocks is included with the design tool software (e.g., as part of a database) so that they may be substituted into the system level netlist of the PLL (e.g., as depicted in Figure 6d) in accordance with methodology 550 of Figure 5b. Likewise, a netlist that describes the topology of the loop filter 703b may also be included with the design tool software. More detail regarding the automatic expansion of the PLL netlist are described below in Section 7.0.

### 6.3 Charge Pump

**[00122]** Figure 8a shows an embodiment of a family of monomial/posynomial equations 820 through 826 that describe the charge pump variables that the PLL system level description embodiment of Figure 6a is dependent upon (as summarized in table 602b of Figure 6b). Here, note that equations 820 and 822 through 826 are posynomial; and, equation 821 is monomial. Figure 8b shows additional equations 802b (in posynomial and/or monomial form) that can be added to the family of equations for geometric problem solving. Here, the equations of Figure 8b describe the “DC biasing” conditions of the charge pump circuit. An embodiment of a charge pump circuit 802c to which both sets of equations 802a, 802b relate is shown in Figure 8c.

**[00123]** According to the circuit design approach of Figure 8c, a current mirror arrangement is established between four current legs I1, I2, I3, I4. Here, M9 acts as the basis for the current mirroring. As such, the current I1 through M9 (and M10) is designed to be equal to: 1) the current I2 through M5a (and M6a and M8a and M7a); 2) the current I3 through M6b (and M8b and M7b); and 3) the current I3 through M5c (and M6c and M8c). Given the current equalities in the four current legs I1 through I4, the equation 820 for the power consumption of the charge

pump (as expressed in Figure 8a) is the product of VDD and the summation of four times the current through M9 (M9.ID) plus the current through M5 (M5.ID). Equation 821 defines the charge pump current CP.IP as the pulling current of the charge pump. Here, note that the syntax “ID” is used to refer to the drain to source current of the identified transistor.

[00124] Equation 822 describes the difference between the pushing charge pump current and the pulling charge pump current (where the pushing charge pump current flows through M7 and M8 and out output 815; and, the pulling charge pump current flows into output 815 and through M6 and M5). Equation 822 describes this quantity as  $VDD/2$  times the output conductance of the charge pump. Here, the output conductance of the charge pump is a function of the conductance of each of transistors M5 through M8 9as indicated by syntax “gd”; and, the transconductance of transistors M6 and M8 (as indicated by syntax “gm”).

[00125] Equation 823 expresses the variance in the charge pump current difference described just above. The variance in charge pump current is a function of the variance of the drain current of each of transistors M5 and M7. Here the first two terms of equation 823 correspond to the drain current variance of transistor M5 while the last two terms of equation 823 correspond to the drain current variance of transistor M7. Here, W refers to transistor gate width, L refers to transistor gate length, NF refers to the number of fingers used to implement the transistor (as such the effective transistor width corresponds to  $W(NF)$ ). DELTA.VT corresponds to the variation in transistor threshold voltage and Z corresponds to the percent variation in the saturation current (which are both process dependent numbers obtainable from the target manufacturing process).

[00126] Equations 824 and 825 relate to the offset and variance associated with the charge that is added to (or removed from) the loop filter via output 815 by means other than the pushing

current circuitry (which is embodied by transistors M7 and M8) and the pulling current circuitry (which is embodied by transistors M6 and M5). Specifically, the charge referred to just above is modeled as arising out of the voltage follower that is constructed with operational amplifier 810. Here, the offset input voltage associated with the amplifier 810 (OPAMP.DELTA\_VIN; and, in units of variance: OPAMP.VARIANCE\_VIN) multiplied by the capacitances associated with transistors M6, M2, and M1; or transistors M8, M4 and M3 is the source of the charge which is transferred to the output 815.

[00127] As such, both of equations 824 and 825 are functions of the opamp 810 offset voltage and these capacitances (where  $C_{gs}$  corresponds to gate to source capacitance,  $C_{gd}$  corresponds to gate to drain capacitance; and  $C_{db}$  corresponds to the drain to bulk capacitance. Equation 826 expresses the area of the charge pump as the summation of the area of each of its constituent transistors.  $\beta$  is a process dependent variable obtainable by those of ordinary skill. Note that the number of fingers (NF) may be taken into account for the transistors (e.g., by incorporating the NF value for the transistor in its corresponding product term).

[00128] Figure 8b shows an embodiment of a family of DC biasing related equations 802b (expressed in posynomial and/or monomial form) that can be added to the family of equations used for geometric problem solving. Adding DC biasing equations to the family of monomial and posynomial equations helps to ensure that the geometric solution (when obtained) will correspond to a solution where the proper biasing of the charge pump is accounted for. Here, monomial equations 827 effectively describe the current relationships between the transistors M5, M6, M7, M8, M1 and M3 responsible for pushing or pulling current to/from the loop filter. The current mirror relationship between  $I_1$  (M9.ID) and current legs  $I_3$  and  $I_4$  (M6b.ID and M8c.ID) is also described.

[00129] Equations 828 describe additional current mirroring relationships. Here, note that VGOV corresponds to the term  $V_{GS}-V_T$  where  $V_{GS}$  is gate to source voltage and  $V_T$  is the threshold voltage. Equations 829 describe conditions that keep the transistors of the charge pump in “saturation” mode. Here, saturation mode corresponds to a state of high output conductance. Here, by keeping the transistors in their saturation region, the transistors will be biased so as to be kept in a region that is appropriate for their implementation.

[00130] Figure 8c shows an embodiment of a topology for the charge to which the equations of Figures 8a and 8b relate. Here, note that the phase detector output is differential for both the “up” and “down” error term signals. That is, a pair of “up” charge pump inputs exist (811, 812) and a pair of “down” charge pump inputs exist (813, 814). This information may be further added to the system level netlist description; or, the system level netlist may be made to account for differential outputs in its initial definition.

#### 6.4 VCO

[00131] Figure 9a shows an embodiment of a family of monomial/posynomial equations 920 through 925 that describe the VCO variables that the PLL system level description embodiment of Figure 6a is dependent upon (as summarized in table 602b of Figure 6b). Here, note that equations 920, 921 and 923, 924 are posynomial while equation 922 and 925 are monomial (noting that the expression  $\mu_n + \mu_p$  is a constant as described in more detail below). Figure 9b shows additional equations 905b (in posynomial and/or monomial form) that can be added to the family of equations for geometric problem solving. Here, the equations of Figure 9b describe the “DC biasing” conditions of the VCO. An embodiment of a VCO circuit 802c to which both sets of equations 902a, 902b relate is shown in Figure 9c.



[00132] According to the design of Figure 9c, a ring of N delay stages (where each delay stage corresponds to a pair of totem poled Mp and Mn transistors (e.g., M1p and M1n correspond to a first delay stage; M2p and M2n correspond to a second delay stage; and, MNp and MNn correspond to the Nth delay stage). The delay stages are arranged in a ring such that an oscillation is created at output 910. The frequency of the oscillation is a function of the voltage on node 911. Here, as the voltage rises on node 911, the frequency of oscillation increases (because the delay stages operate “faster”); and, likewise, as the voltage on node 911 falls, the frequency of oscillation decreases. As the voltage on node 911 is correlated to the voltage presented at input node 912, the voltage to frequency conversion performed by the VCO is readily apparent.

[00133] Referring to Figure 9a, the power consumption of the VCO is expressed by equation 920 as the power consumption of opamp 913 and the product of Vdd and the summation of the current through M21 (M21.ID) and NMx1.ID. Here, NMx1.ID corresponds to the current through each of transistors Mx1 through MxN. The area of the VCO is expressed by equation 921 as the summation of the areas of its constituent transistors and the area of the opamp 912. The gain of the VCO is expressed by equation 922 as the product of: 1) the change in output signal frequency per change in current provided to the delay stages; and 2) the change in current provided to the delay stages in response to a change in voltage at input node 912.

[00134] Here, the first bracketed term in equation 922 correspond to the first relationship and the second bracketed term in equation 922 corresponds to the second relationship. Note that the quantity  $\mu_n + \mu_p$  corresponds to the summation of electron mobility and hole mobility in semiconductor materials and may therefore be viewed as a constant. Similarly,  $C_{ox}(1+ratio)$  also corresponds to a constant for a particular semiconductor manufacturing process (where  $C_{ox}$

is the metal-oxide-semiconductor capacitance and ratio is set to 3.0 in one embodiment (which corresponds to an approximation undertaken to place equation 923 in monomial form).

**[00135]** N corresponds to the number of delay stages in the inverter and is a variable to be solved by the geometric solving process. Mn.W, Mn.ID and Mn.L correspond to the gate width, current and gate length of one of the n type delay stage transistors. Equation 923 describes the VCO.KAPPA term with similar variables as well as boltzman's constant "k", the temperature T, the transconductance of both an Mn and Mp type transistor within a delay stage (Mn.gm and Mn.gm), the VCO output frequency; and, the gate width of a p type transistor within a delay stage (Mp.W). The VCO power supply rejection VCO.VDD\_DC is expressed with terms mentioned above as well as the opamp gain (OPAMP.GAIN) and the resistance rc between the + and - inputs of the opamp.

**[00136]** The VCO.TAU3 parameter is described in terms of the opamp input transconductance (OPAMP.gm\_in) as well as the gate to source capacitance of M24 (M24.Cgs), the transconductance of transistor M22 (M22.gm) and the conductance of transistor M22 (M22.gd). Figure 9b shows DC biasing conditions in terms of current mirrors 926, 927 expressed in monomial form as well as dimensional relationships between the transistors within the delay stages that are also expressed in monomial form. Conditions for keeping the transistors operating velocity saturation are also provided in posynomial equations 924 and 930.

## 6.5 Comment About Transistor Level of Detail

**[00137]** Note that the present section, which has examined the PLL's building blocks at a greater level of detail, has revealed transistor level information for many of these building blocks. For example, the circuit topology of the charge pump (as observed in Figure 8c) reveals

20 different transistors and an op-amp 810. Likewise, the family of posynomial and monomial equations for the charge pump (as observed in Figures 8a and 8b) are replete with transistor level details. That is, variables such as drain to source current ( $I_D$ ), gate to source capacitance ( $C_{gs}$ ), gate to drain capacitance ( $C_{gd}$ ), transconductance ( $g_m$ ), gate length ( $L$ ), gatewidth ( $W$ ) are observed throughout Figures 8a and 8b. Similar comments may be said about the VCO as well.

[00138] Here, it is important to note that although a geometric problem solving sequence could be attempted at the current level of detail; in alternate embodiments, various transistor level characteristics (e.g.,  $C_{gs}$ ,  $C_{gd}$ ,  $g_m$ , etc.) could be even further refined so as to be described in terms of the gates widths and gate lengths of their specific transistor. Here, equations from standard textbooks could be used (either directly because the relationships are normally expressed in monomial or posynomial form; or, perhaps some approximations may be undertaken to craft a relationship in posynomial or monomial form) Regardless, a geometric problem can be solved where most the variables correspond to the gate widths and gate lengths of the various transistors that take part in the optimization process. As a result, the geometric solution will provide specific dimensions for the transistors that are characterized in this fashion.

## 7.0 Automated Design Tool Technology

[00139] Figure 10 shows an embodiment of a methodology 100 by which an automated design tool designed according to the principles expressed above may be configured to operate according to. Initially a system level description of a PLL is created 1001. Recall that an example of a system level description was discussed with respect to Figures 6a through 6e. Here, the construction of a system level description results in system level topology information

1006 (e.g., the netlist of Figure 6d) and system level monomial and/or posynomial equations 1007 (e.g., a example of which were shown in Figure 6a).

**[00140]** Once the system level description has been created 1001, lower level circuit topology information and monomial and posynomial expressions are substituted, respectively, into higher level circuit topology information and posynomial expressions until a transistor level circuit topology is formed. In an embodiment, the equations are continually substituted until a family of monomial/posynomial expressions are developed having mostly undefined transistor dimensions 1002. Embodiments of methodologies that may be used to perform the substations 1002 of Figure 10 have been shown with respect to Figures 5a and 5b (and, where, the word “system” may be replaced with the phrase” “basic analog building block” as needed in each of these figures). The software may make use of database 1003 that stores the information to be substituted into the higher level descriptions.

**[00141]** Once the transistor level has been reached, transistor level circuit topology information 1008 (e.g., having undefined transistor dimensions )results; and, a family of monomial and posynomial equations 1009 that express the PLL at the transistor level is created. Then, a geometric problem is constructed and solved 1004 which provides numeric details 1010 that define the previously undefined transistor parameters (e.g., dimensions). A particular objective (such as minimizing a particular characteristic such as power consumption or surface are consumption) is usually articulated when constructing the geometric problem. Then, for example, the defined transistor dimensions 1010 are combined 1005 with the transistor level circuit topology information (having undefined transistor level dimensions) so as to form a more complete transistor level description 1011 of the circuit.

[00142] Figure 11 shows an embodiment as to how the system level description of the PLL may be formed 1001. According to the approach of Figure 11, definitional information of a system is created or received or otherwise provided 1110. Here, definitional information effectively defines the “type” of circuit or system to be constructed (e.g., PLL in this case). Then, system level circuit topology information and system level monomial and/or posynomial expressions are retrieved or constructed 1111. This may be done by the designer, by the software automatically (e.g., by receiving a generic data file from the database for a PLL) or some combination of both. This produces a system level circuit topology 1106 (e.g., a system level netlist for a PLL) and a system level family of monomial and/or posynomial expressions that are left unconstrained (e.g., without specific numeric constraints). Then, constraints 1115 are applied 1112 to the family of unconstrained equations 1114 (e.g., in the form of user\_specs). So as to update the family of equations with the desired constraints. Note that each and every equation in the family of equations need not receive a specific constraint (indicating the designer is allowing the particular characteristic to vary at will with the geometric problem solving process).

[00143] Figure 12 provide a specific example as to how the PLL’s family of equations become more detailed with each substitution. Figure 12 shows the original system level equation for PLL power consumption (as expressed in Figure 6a). Equation 1201 of Figure 12, shows this same equation after the CP.PWR term is substituted with equation 820 of Figure 8a and the VCO.PWR term is substituted with equation 920 of Figure 9. Here, note that equation 1201 effectively describes the PLL’s power consumption in greater detail by expanding on each of the CP.PWR and VCO.PWR terms with transistor level expressions. Likewise, for each of the non

user defined variables of Figure 6b, a similar substitution could be made which would result in the entire PLL being described at a greater level of detail.

**[00144]** Figure 13 shows an example as to how the system level netlist can be made to be more detailed as lower level circuit topology information is substituted into the system level netlist. Figure 13 shows the simple case of the changes made to the system level netlist of Figure 6d if the circuit topology information of the loop filter of Figure 7b is substituted in. Here, note that nodes 6 and 7 of the system level netlist of Figure 6d are merged because the loop filter topology of Figure 7b indicates that the input and output of the loop filter are the same node. As such, whereas the CP.OUT and VCO.IN nodes were on different nets in the system level description of Figure 6D, they now share the same node in the more detailed netlist of Figure 13. Furthermore, whereas the system level netlist of Figure 6d is devoid of specific resistor and capacitor values (which are considered a low level of detail), note that this information appears when the loop filter specifics are accounted for. Specifically, nodes 2, 6 and 7 contains specific resistance and capacitance details (noting that node 7 was reconfigured so as to correspond to the node between R and C1). Similar to the discussion above with respect to Figure 12, each insertion of more detailed topological information (e.g., the design for the charge pump or VCO as provided in Figures 8c and 9c) will add more nodes and more complexity to the netlist (which corresponds to the netlist being described at a greater level of detail).

**[00145]** Figure 14 shows the various threading paths and threading lengths that may arise as a PLL is automatically developed by continuously substituting lower levels of detail into higher levels of detail. Figure 14 shows how the both the charge pump and vco can be expanded into a transistor level of detail. Here, as observed in Figures 8c and 9c both the charge pump and VCO are described in terms of an opamp and a plurality of transistors (20 transistors in the case of the

charge pump and  $4 + 3N$  transistors in the case of the VCO). As such a first substitution 1401 (for both the charge pump and VCO) will bring the PLL closer to a transistor level of detail by  $24 + 3N$  transistors. However, because of the opamps, both the charge pump and VCO will not be broken down completely into transistor level terms until a second substitution 1402 is made that effectively substitutes transistor level information for each of the opamps (e.g., as described with respect to Figures 3a and 3b).

**[00146]** Figure 15 shows an embodiment of a computing system 1500 that can execute instructions residing on a machine readable medium (noting that other (e.g., more elaborate) computing system embodiments are possible). The instructions may be related to integrated circuit design (e.g., as described in Figures 2a through 15). In one embodiment, the machine readable medium may be a fixed medium such as a hard disk drive 1502. In other embodiments, the machine readable medium may be movable such as a CD ROM 1503, a compact disc, a magnetic tape, etc. The instructions (or portions thereof) that are stored on the machine readable medium are loaded into memory (e.g., a Random Access Memory (RAM)) 1505; and, the processing core 1506 (e.g., a central processing unit comprising one or more processors) then executes the instructions. The instructions may also be received through a network interface 1507 prior to their being loaded into memory 1505.

**[00147]** Thus, it is also to be understood that embodiments of this invention may be used as or to support a software program executed upon some form of processing core (such as the Central Processing Unit (CPU) of a computer) or otherwise implemented or realized upon or within a machine readable medium. A machine readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine readable medium includes read only memory (ROM); random access memory (RAM);

magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

**[00148]** In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.